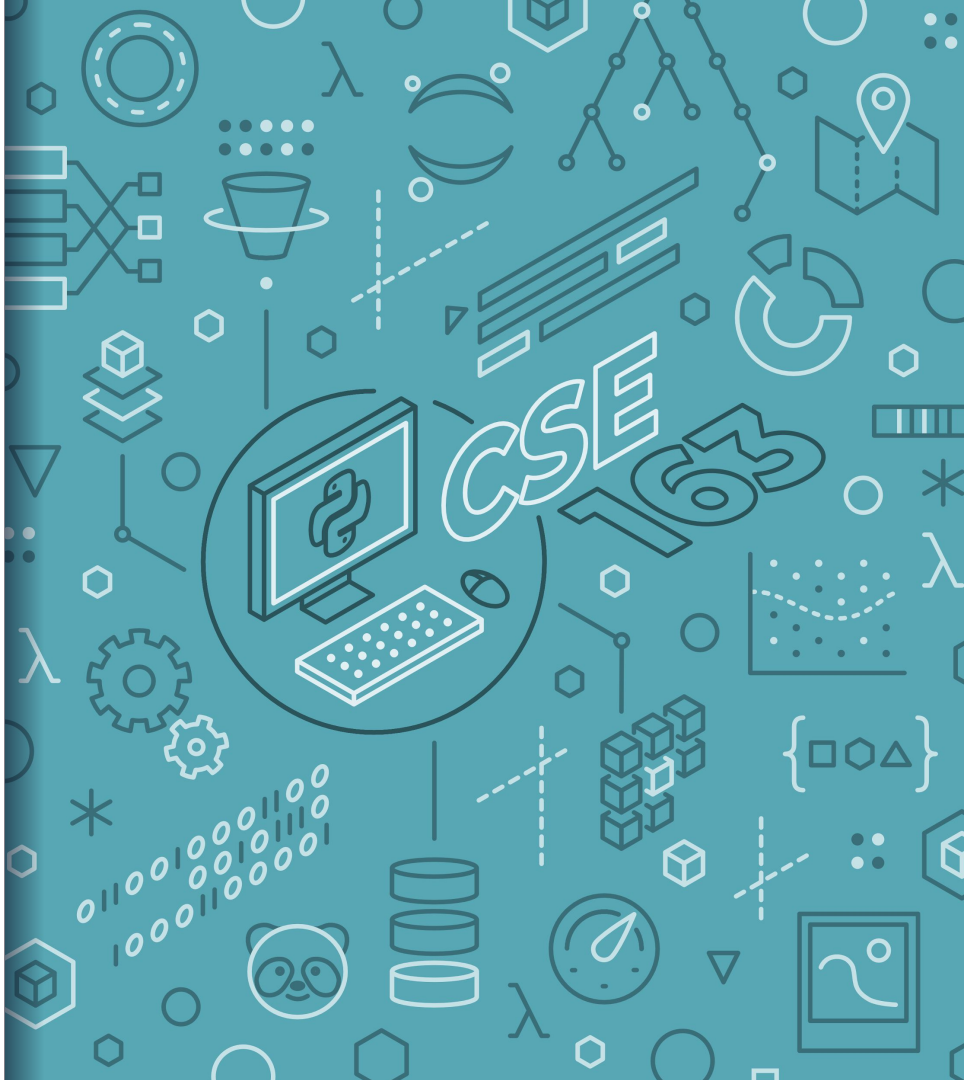
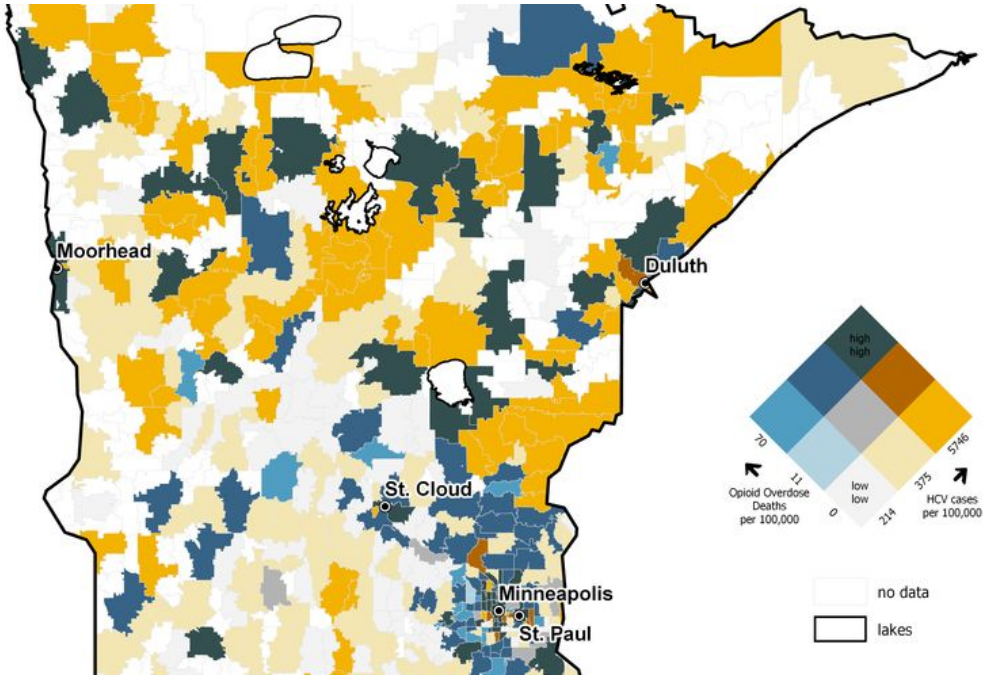
CSE 163

Geospatial Data / Geopandas

Hunter Schafer



Geospatial Data



Geospatial Data

- A lot of data exists in the “real world”
  - It’s not just a data point, but it’s related to a place or group of people
- Geospatial data lets you process your data not only by their values, but also by its location in the world

Geopandas

- Exactly like pandas, but with some extra features for geo data
- Types
  - GeoDataFrame
  - GeoSeries

```
data = geopandas.read_file('data_file.shp')
```

- Many types of files
  - Shapefile (.shp and friends)
  - GeoJSON (.json/.geojson)
  - GeoPackage (.gpkg)
  - Many more!

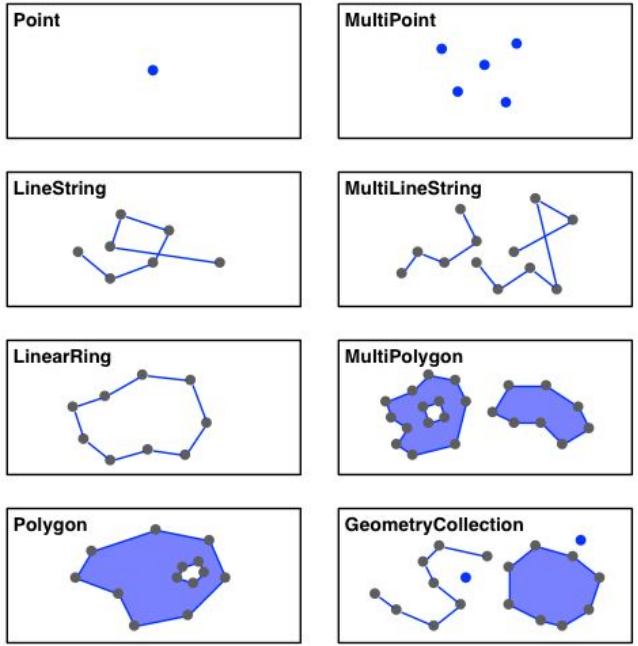
Geopandas

- Read a file containing information about countries in the world
  - E.g. population, GDP
- Make plots to show geospatial relation of GDP or population
- Can filter just like a regular DataFrame!

Colab

Geometry

- One very common notion in geospatial data is a **geometry**
- This is what actually defines the shape of the countries in the last example
- Types of geometry: `shapely.geometry`



Brain Break

Hurricane Florence

Here’s the trick: This information comes from 2 different datasets!

Florence Example

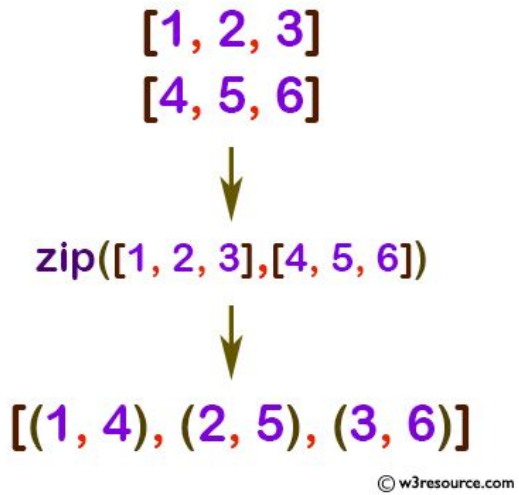
- Parse data into multiple DataFrames
- Plot each dataset over each other

Colab

zip

- **zip** is a helpful built-in function to help you combine lists

```
x = [1,2,3]
y = [4,5,6]
zip(x, y)
```



- Note: zip does not return a list! It returns a **generator**
  - Can only access the values by iterating over them
  - Can only be used once! Iterating “consumes” the series

Matplotlib

- Before, we were using a lot of matplotlib stuff blindly
- Default plots gets plotted on a global **figure** (think canvas)
- Every time you make a new plot, it will be painted on the same figure, but on a different set of **axes**
- You have to do extra work to get more information on the same plot

```
fig, ax = plt.subplots(1)
<plot1>(ax=ax)
<plot2>(ax=ax)
fig.show()
```